



SANDIA REPORT

SAND2001-1780

Unlimited Release

Printed September, 2001

SLINGSHOT – a Coilgun Design Code

Barry Marder

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.doe.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/ordering.htm>



SAND2001-1780
Unlimited Release
Printed September, 2001

SLINGSHOT- a Coilgun Design Code

Barry Marder
Dept 1674, Target and z-pinch theory
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185-1186

Abstract

The Sandia coilgun is an inductive electromagnetic launcher. The tool most widely used for modeling coilgun behavior is the SLINGSHOT computer code. This report describes the code and its use. It is assumed that the reader is familiar with coilguns.

Contents

I. Introduction	3
II. A Benchmark Example	3
III. Using SLINGSHOT	6
a) Input	6
b) Output	11
IV. Physical Model, Numerical Methods	12
a) Circuit equations	12
b) Inductance calculations	13
Acknowledgements	15
References	15

Figures

1. Coil circuit	4
2. Benchmark geometry	4
3. Benchmark dynamics	5
4. Benchmark energy	5
5. Benchmark currents and forces	5
6. Alternative armature	9
7. Lyle's method, current filaments	14

SLINGSHOT- a Coilgun Design Code

I. Introduction

The Sandia coilgun [1,2,3,4,5] is an inductive electromagnetic launcher. It consists of a sequence of powered, multi-turn coils surrounding a flyway of circular cross-section through which a conducting armature passes. When the armature is properly positioned with respect to a coil, a charged capacitor is switched into the coil circuit. The rising coil currents induce a current in the armature, producing a repulsive accelerating force.

The basic numerical tool for modeling the coilgun is the SLINGSHOT code, an expanded, user-friendly successor to WARP-10 [6]. SLINGSHOT computes the currents in the coils and armature, finds the forces produced by those currents, and moves the armature through the array of coils. In this approach, the cylindrically symmetric coils and armature are subdivided into concentric hoops with rectangular cross-section, in each of which the current is assumed to be uniform. The ensemble of hoops are treated as coupled circuits. The specific heats and resistivities of the hoops are found as functions of temperature and used to determine the resistive heating. The code calculates the resistances and inductances for all hoops, and the mutual inductances for all hoop pairs. Using these, it computes the hoop currents from their circuit equations, finds the forces from the products of these currents and the mutual inductance gradient, and moves the armature. Treating the problem as a set of coupled circuits is a fast and accurate approach compared to solving the field equations. Its use, however, is restricted to problems in which the symmetry dictates the current paths.

This paper is divided into three parts. The first presents a demonstration of the code. The second describes the input and output. The third part describes the physical models and numerical methods used in the code. It is assumed that the reader is familiar with coilguns.

II. A Benchmark Example

A simple benchmark problem is used to illustrate the code's functioning and capabilities. The armature is a hollow aluminum cylinder with outer radius 2.5 cm, inner radius 1 cm, and length 10 cm. There are forty 20-turn coils, each with an inner radius of 3 cm, an outer radius of 4.5 cm, an axial width of 0.25 cm, and a center-to-center spacing of 3.5 cm. The coils are connected to an external circuit and coupled to the armature, as shown in Figure 1. The 150 μF capacitors are charged to 20 kV. The resistances R_a , R_b , and R_d are all 10 m Ω . The corresponding inductances, L_a , L_b , and L_d , are all 100 nH. The first 10 coils are "crowbarred" by closing Switch 2 when the capacitor has discharged; the remainder are allowed to ring. The "slip", the rate at which the firing position on the armature advances, is 3 m/s. The resistance and inductance of the coils, R_c and L_c , and the armature (or projectile), R_p and L_p , are calculated by the code from the geometry, material, and temperature. The code offers several algorithms to determine when to close Switch 1.

Figure 2 shows the geometry and magnetic field after the armature has moved about 12 cm. The armature is approximated as a collection of rectangular cross-section hoops whose number and geometry are chosen to provide sufficient resolution of the current density distribution. Because the current density is uniform in any hoop, the outer surface elements are thinner to better model surface penetration on the skin-depth scale. The multi-turn coils are modeled as single elements because their current density is uniform across the face. Figure 3 shows the velocity and acceleration; Figure 4 gives the energy accounting; and Figure 5 gives the currents and forces for this benchmark SLINGSHOT run.

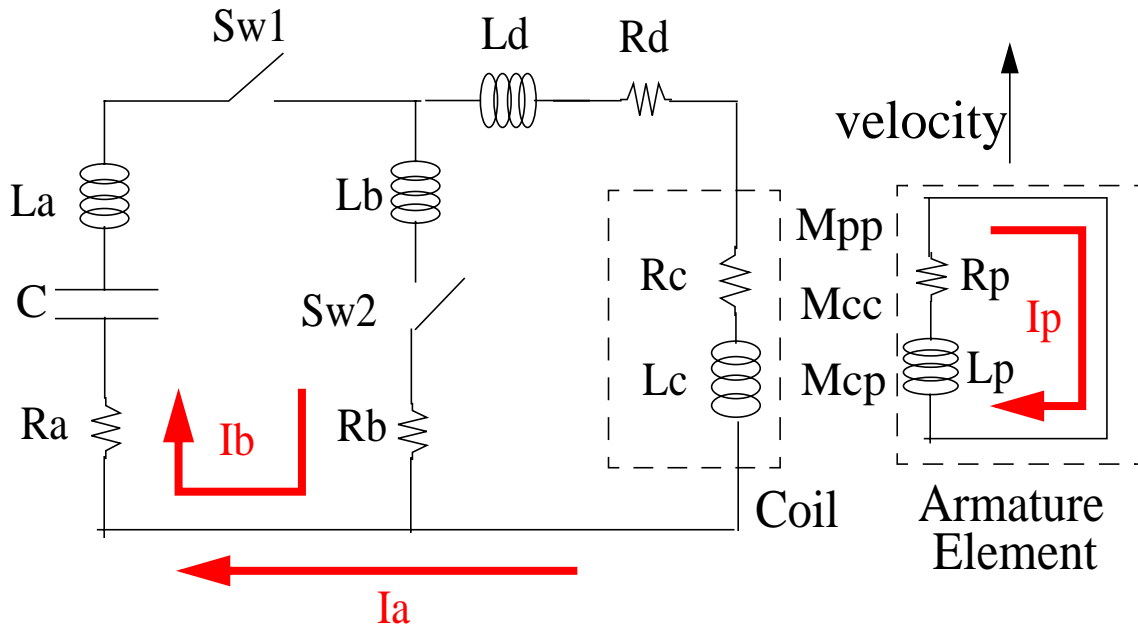


Figure 1. Circuit for a single coil and representative circuit for the (possibly) multi-element armature. All circuits, coils and armature, are mutually coupled.

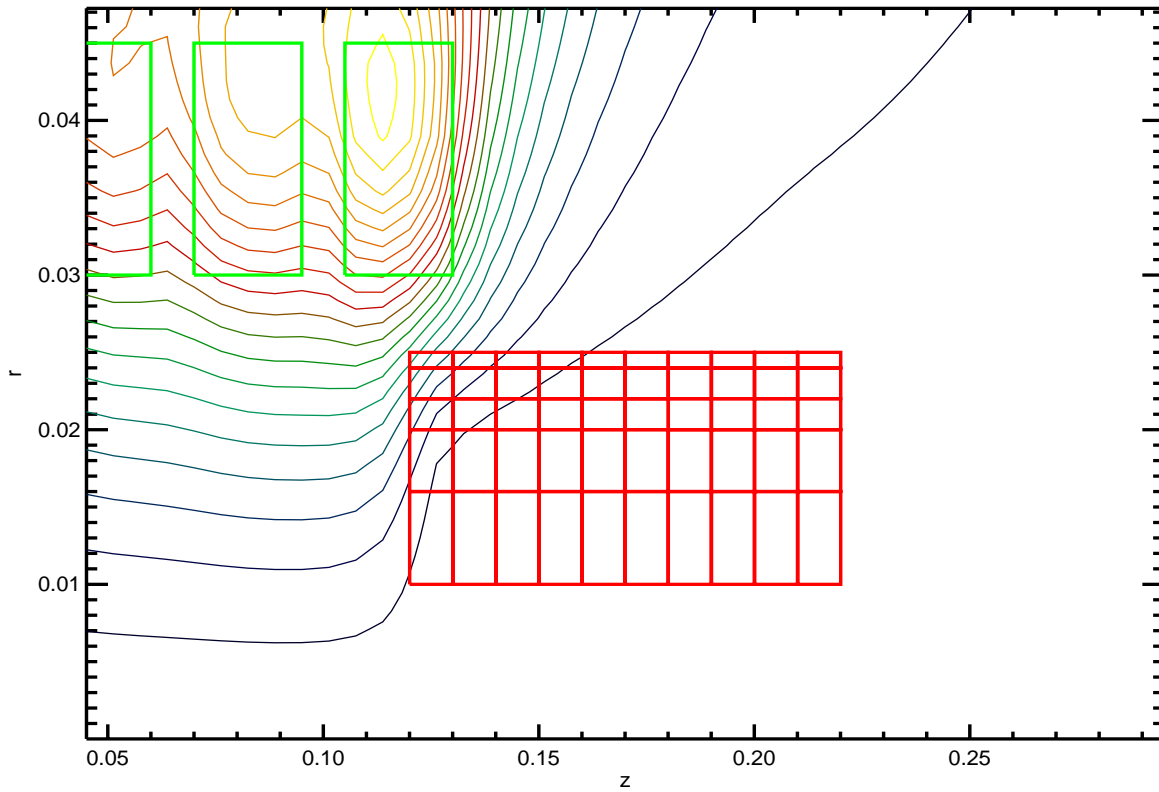


Figure 2. Benchmark geometry showing the coils (green), armature (red), and magnetic field contours. Only coils that have fired are shown. $z=0$ is the start of the first coil.

Figure 3. Acceleration, velocity, and peak armature temperature vs. distance along the gun.

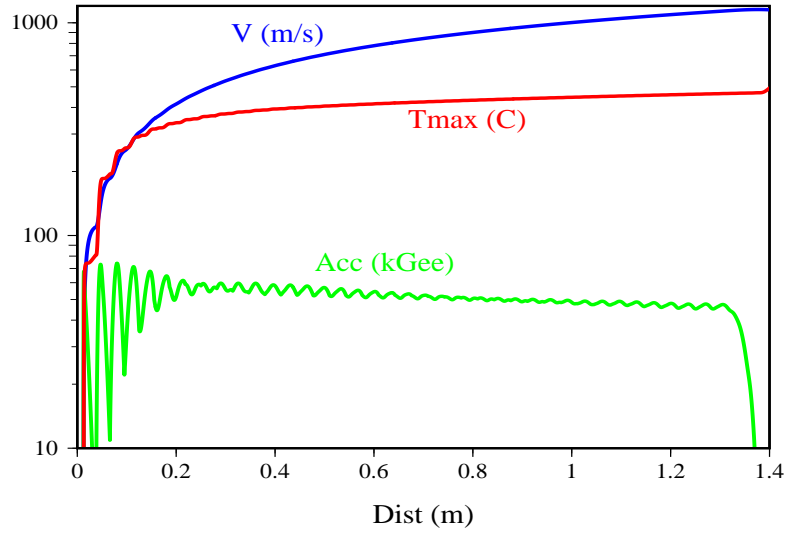


Figure 4. Energy accounting: coil (R_c) and armature (R_a) heat, inductive (L), capacitive (C), kinetic (K). The sum of the above and the integrated input energy (should be equal).

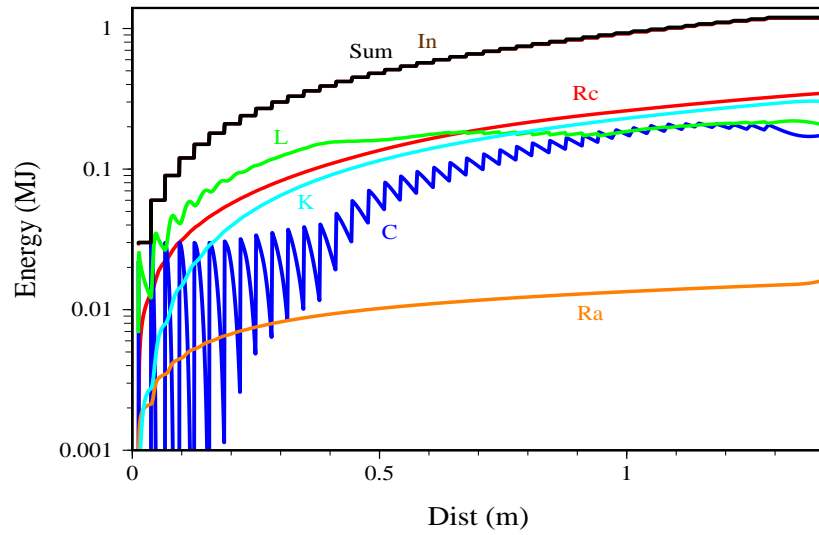
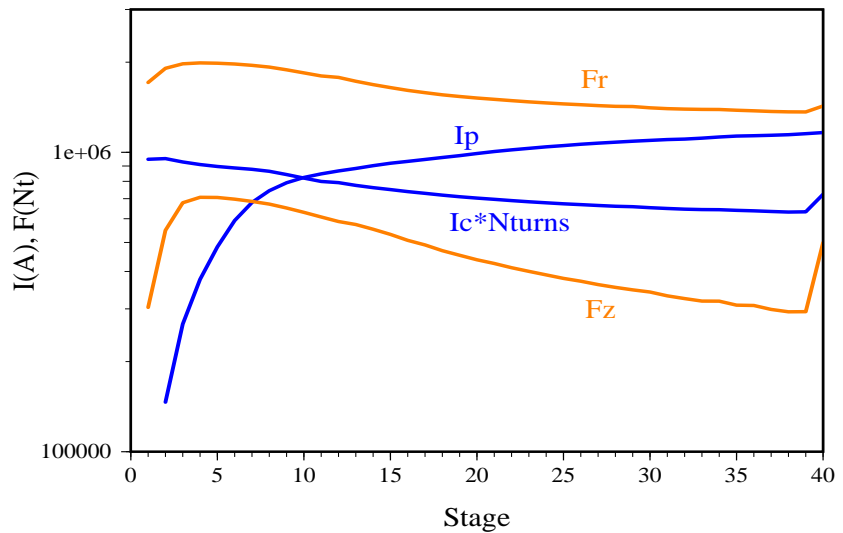


Figure 5. Armature current (I_p) and the number of turns times the peak coil currents (I_c) vs. the stage number. Also plotted are the axial and radial forces on the coils.



III. Using SLINGSHOT

a) Input

SLINGSHOT is written in FORTRAN 77 and is self-contained. It is designed to be run interactively. It reads an input file called *fort.20*. The input file for the benchmark is:

```
'1. Input file for slingshot, 40 stage SAND report benchmark.'
```

```
'2. 0: Starting from t=0. 1: continuation, reading file 29 (irun)'
```

```
0
```

```
'3. a) Pause when the rear edge of the armature gets here (zstop)'
```

```
'  b) Time at which code will stop regardless of position (tstop)'
```

```
1.4 .01
```

```
'4. Number of elements: a) armature b) extra c) coils'
```

```
50 0 40
```

```
'5. Number of coils behind armature used in calculation.'
```

```
20
```

```
'6. Armature elements:'
```

```
' 0: index(1-narm), material, r1,r2,z1,z2 (z1 must be 0. Code will add initial position, z0)'
```

```
' 1: material, ncol, nrow, columns (ncol+1), rows (nrow+1) [from top!]'
```

```
1
```

```
11 10 5
```

```
0. 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10
```

```
0.025 0.024 0.022 0.020 0.016 0.010
```

```
'7. Extra elements'
```

```
' index(1-nextra), material, r1,r2,z1,z2'
```

```
'8. Coil elements:'
```

```
' 0: index(1-ncoils), material, r1,r2,z1,z2 (z1 of the first coil must be 0)'
```

```
' 1: material, inner and outer radii, width, periodicity'
```

```
1
```

```
20 .03 .045 .025 .035
```

```
'9. a) Payload mass (excluding armature)'
```

```
'  b) Initial position of back edge of armature relative to center of first coil,'
```

```
'  c) Initial velocity'
```

```
'  d) Slip. Firing position advances with slip*time or |slip|*dist (slip<0))'
```

```
'  e) Rise length (needed when nturns = 0)'
```

```
'  f) Rear of armature relative to coil center at peak current'
```

```
' 0. 0. 0. 3. 0. 0.'
```


'10. Number of turns for each of the coils (1-ncoils)'
 ' If nturns = 0, code determines value based on given rise length'
 40*20

'11. Minimum number of coil turns'
 2

'12. 0: Ringing, 1: crowbar, 2: open 1 at T/2, 3: at T, 4: sw1 open (1-ncoils)'
 10*1 30*0

'13. Charge voltages for each of the coils (1-ncoils)'
 40*20.e3

'14. Capacitance Ca for the coil circuits (1-ncoils)'
 40*150.e-6

'15. Resistances Ra, Rb, and Rd for the coil circuits (1-ncoils)'
 40*.01 40*.01 40*.01

'16. Inductances La, Lb, and Ld for the coil circuits (1-ncoils)'
 40*100.e-9 40*100.e-9 40*100.e-9

'17. Volume fraction filled by conductor for all elements (1-narm)&(1-nextra)&(1-ncoils) '
 50*1. 40*0.75

'18. Initial temperature (degrees C) for all elements (1-narm)&(1-nextra)&(1-ncoils)'
 50*-196. 40*20.

'19. 0: firing locations computed'
 ' 1: dist of coil centers from back of armature at firing (1-ncoils)'
 ' 2: times of firing (1-ncoils)'
 0

'20. The number of time steps between writes to files 26 and 27.'
 ' If -1, file written only once, at end of run.'
 -1

'21. Factor which multiplies the time step calculated by the code'
 1.

The input file must follow this template, with each line of text enclosed in single quotes. The wording is not important; the text is there only to identify and separate the input. The identifying text line numbers could, but need not, be included in an actual input file. The input parameters are printed in red. All units are MKS; the temperature is Centigrade. We shall consider each line of input by number.

1. The first line identifies the file.

2. The code can be stopped at any time and restarted. The output file *fort.28* contains the final state. To continue, rename (or copy) the file to *fort.29* and set *irun*=1.

3. The run will pause when the rear of the armature reaches zstop. It will query the user whether to stop or continue. It will always stop if the time exceeds tstop. This limits the run time if the velocity stays small and the armature would take a very long time to reach zstop. The code automatically stops if the velocity becomes negative.

4. The number of elements in the armature, the number of extra elements, and the number of coils is given. The extra elements are simply loops which neither move nor are connected to an external circuit. They are included, for example, to allow support rings or other structures in which currents can be induced and which could, therefore, extract energy or exert forces.

5. Once the armature is well past a coil, it is no longer necessary to include that coil in the calculation. This parameter specifies how many active coils are in the “tail” behind the armature. It must be large enough so that coils behind the tail couple weakly to both the armature and other stator coils which are still interacting with the armature. The smaller the tail, the faster the calculation will run.

6. The armature material and shape are defined here. Only relative z values matter. The code will move the armature to the starting location defined in input 9. There are two options to define the coaxial, single turn elements of the armature. The second one, designated by 1, is used for a single material, cylindrical armature partitioned into a rectangular mesh, as in the example. The axial divisions (columns) and radial divisions (rows) of the cell boundaries are given, starting from the outer radius. Notice that there is one more boundary than there are cells (one cell has two boundaries). SLINGSHOT has the capability to handle several materials. These are identified in the subroutines which define them:

10-19 is aluminum (10=common, 11=7075 alloy, 12&13=al-fe-ce alloys)
 20-29 is copper (20=common, 21=c17510 alloy, 22=be cu)
 30-39 is tungsten (30=common)
 40-49 is iron & steels (40=iron, 41=maraging steel, 42 timken MP35N)
 50-59 is carbon (50=common)
 60-69 is titanium (50=Ti6Al4V)

The material used in the benchmark for the armature is number 11, aluminum alloy 7075.

The other option for specifying the armature elements is designated by 0. For this case, every element is individually identified, both by physical location and material. This is used for armatures which are cylinders whose cross-section is not rectangular or which contain multiple materials. As an example, input 6 for a thinner armature with a tungsten (material 30) plug, shown in Figure 6, is:

```

0
1  11 .022 .025 0. .004
2  11 .022 .025 .004 .009
3  11 .022 .025 .009 .014
4  11 .022 .025 .014 .020
5  11 .022 .025 .020 .026
6  11 .022 .025 .026 .032
7  11 .022 .025 .032 .038
8  11 .022 .025 .038 .044
9  11 .022 .025 .044 .050
10 11 .022 .025 .050 .056

```

11	11	.022	.025	.056	.062
12	11	.022	.025	.062	.068
13	11	.022	.025	.068	.074
14	11	.022	.025	.074	.080
15	11	.022	.025	.080	.086
16	11	.017	.022	0.	.004
17	11	.017	.022	.004	.009
18	11	.017	.022	.009	.014
19	11	.017	.022	.014	.020
20	11	.017	.022	.020	.026
21	11	.017	.022	.026	.032
22	11	.017	.022	.032	.038
23	11	.017	.022	.038	.044
24	11	.017	.022	.044	.050
25	11	.017	.022	.050	.056
26	11	.017	.022	.056	.062
27	11	.017	.022	.062	.068
28	11	.017	.022	.068	.074
29	11	.017	.022	.074	.080
30	11	.017	.022	.080	.086
31	30	0.	.010	0.	.010
32	30	0.	.010	.010	.020
33	30	.010	.017	0.	.010
34	30	.010	.017	.010	.020

Because there are 34 elements in this armature, inputs 4, 17, and 18 were also changed.

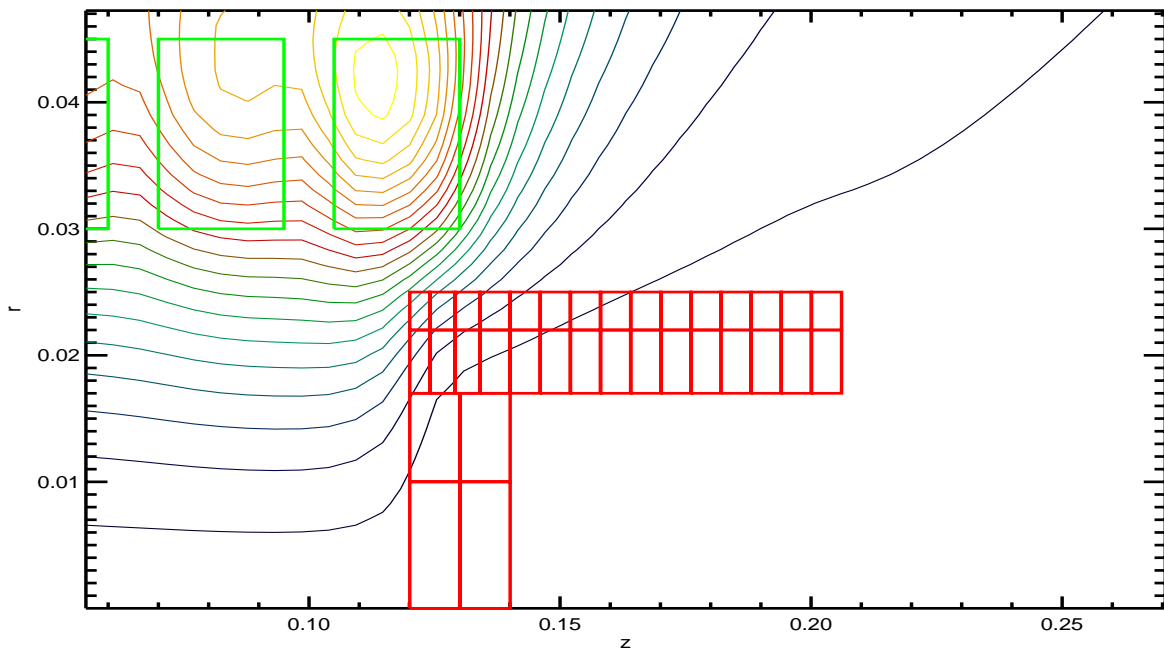


Figure 6. An alternate armature requiring the “0” specification in input 6.

7. If there are extra elements, they should be entered here in a manner similar to the “0” option above. In the absence of extra elements, nothing should be entered.

8. The coil materials and locations are specified. The edge of the first coil with the minimum z value is considered by the code to be $z = 0$ and should be entered accordingly. Again, the “0” option allows them to be entered individually. The simpler “1” option, used in the benchmark, can be used for identical, uniformly spaced coils. Their size and spacing (periodicity) are given. Notice that the coil material, 20, is copper.

9. This defines the launch package. a) The armature mass is calculated by the code, based on the material and geometry. A payload can be added to define the full launch package. b) The initial position of the rear of the armature relative to the center of the first coil. c) The initial velocity. d) The “slip” is the velocity at which the location of peak coil current advances in time along the armature. In the example, it advances by 3 m/s as the armature progresses down the gun. If the slip is too large, this position will move past the front of the armature. To avoid this, a second option can be used. If this parameter is negative, the peak current position is advanced along the armature by an amount proportional to the distance travelled: $\text{position} = |\text{slip}| * \text{distance}$. For example, if the armature is 10 cm long and the gun is 10 meters long, a slip specification of -0.01 would advance the firing position along the armature from 0 at the breech to 10 cm ($0.01 * 10 \text{ m}$) at the muzzle. This option is typically used for long guns where it aids in determining the optimum armature length. e) The time required for the current to reach its peak value is estimated using the circuit parameters. A correction is made to the inductance to account for the volume occupied by the armature. The rise length is the distance the armature moves during the current rise time. A coil is energized when the rear of the armature (plus slip plus specified advance) lies a rise length behind the coil center. If the number of turns is specified, the rise length is already determined so this input is ignored. The code, however, has the capability (input 10) to determine the number of turns needed in the next coil to give the specified rise length. f) Usually the peak current is desired when the rear of the armature (plus the slip) lies under the coil center. If another location is desired, the firing position can be “advanced” by this parameter.

The armatures we have considered were assumed to be made of solid metal. For some applications, a wound wire armature is desirable. In this case, the armature, like the coils, is composed of only one element since the current density is assumed to be uniform across the entire cross-section. Furthermore, slip is not needed so it should be set to zero.

10. This gives the number of turns for each coil. The coil inductance and resistance vary as the square of this parameter. If this input is 0, the code will determine how many turns are required to give the rise length specified in input 9.

11. If the code calculates the number of turns (“0” in input 10), it can calculate a number less than one. This input prohibits that by specifying the minimum allowed.

12. When Switch 1 is closed, the result is basically an LRC circuit, which can oscillate. This input provides the option to: (0) do nothing and let the circuit ring, (1) crowbar it by closing Switch 2 when the capacitor voltage passes through zero, (2) open Switch 1 at the current null after half a period, (3) open Switch 1 after a full period, or (4) never close Switch 1 so the coil is effectively removed from the gun.

13. The charge voltages for all the capacitors are specified.

14. The capacitances for all the coil circuits are given.

15. The three circuit resistances, R_a , R_b , and R_d , are given.

16. The three circuit inductances, L_a , L_b , and L_d , are given.

17. Each loop is treated as a current-carrying circuit whose resistance depends on its rectangular cross-sectional area. This area need not be completely filled with conductor. In multi-turn coils, for example, some of the space will be taken up by wire insulation. To account for this, a fill fraction is specified for each element. In this example, the armature is solid metal, but the copper fills only 75% of the coils' area.

18. Each element is given an initial temperature. Here, the coils are at room temperature (20° C), while the armature is cooled in liquid nitrogen (-196° C) to decrease its resistance.

19. Normally, the code determines when each coil fires (0), as explained above in the paragraph labeled 9. The code also allows either the firing positions (1) or the firing times (2) to be pre-specified. The firing position is the distance from the back edge of the armature to that coil center. When using 1 or 2, values for each coil must be given.

20. Detailed output information will be written on *fort.26* and *fort.27* after every specified number of time steps. These will be written only at the end of the run if this parameter is -1.

21. The code computes a time step based on the circuitry and kinematics. This can be multiplied it by a factor to give either more resolution (< 1) or to make the code run faster (> 1).

b) Output

While SLINGSHOT is running, it displays the following information on the screen: the first coil in the tail, the last coil to fire, the time, the position of the rear of the armature relative to the midpoint of the first coil, the velocity, the acceleration (in kGee, 1000 times the acceleration of gravity), the peak temperature in any armature element, and the number of turns of the last energized coil. A section of the benchmark screen output is:

```
14 33 2.20 msec 1.077 m 1037.95 m/s 47.13 kG 451.38 C n= 20
14 33 2.21 msec 1.077 m 1038.34 m/s 47.11 kG 451.44 C n= 20
15 34 2.21 msec 1.078 m 1038.73 m/s 47.11 kG 451.48 C n= 20
```

When the armature reaches the specified location, the following will appear on the screen:

```
The armature is now at 1.40 meters
Where should it be at the next pause? (type 0 to stop)
```

Typing 0 will end the run. Giving a number greater than 1.40 meters will cause it to continue until it reaches that location, at which time the message will be repeated. If the temperature of any element exceeds its melting point, a message to that effect will be printed at the end of the run. When the code stops it generates several output files and writes the following on the screen:

The input/output files for SLINGSHOT are:

20: Input file for slingshot. Gives defining parameters.

21: STAGE #, Nturns, zp, vp, ap(kGee), Tpmax, Ip
tfire, trise, trise(est), zfire, zpeak

R, L, C, V, Ipeak, Fz, Fr, Tcmax, eff, Pr(kbar) [22]

22: Dynamics: t(ms), last, zp, vp, ap(kG), Ip, Ic, Tpmax [8]

23: Energy accounting: t(ms), zp, rc, cc, lc, mij, rp,
kinetic, tail, sum, input (sum = input) [11]

26: t(ms), zp, itaila, itailb,
 i, Ia(i), Ib(i), V(i), i=1-ixlast, itaila-itailb.
27: t(ms), zp, itaila, itailb,
 i, fz(i), fr(i), tempctg(i), i=1-ixlast, itaila-itailb.
 (26 and 27 written every nwrite steps, and at end)
28: Contains the final state. Rename 29 to restart.
29: Input for restarting slingshot (irun=1)
30: Input for the field contour post processor slingplot.
40: Input for the armature stress code slingstress.

fort.21 describes the conditions when the armature reaches each coil stage. It contains the stage number, the number of turns, the armature location, the velocity, the acceleration, the maximum armature temperature, the total armature current, the time Switch 1 was closed, the actual rise time to peak current, the estimated rise time calculated by the code, the position at which Switch 1 was closed, and the position of peak current, both measured from the rear of the armature to that coil's center. The file also contains the stage coil resistance, coil inductance, capacitance, its peak current, the peak axial and radial forces on the coils, the coil temperature, the efficiency at which the capacitor energy was converted into kinetic (the change in armature kinetic energy during the time step divided by the change in capacitor energy), and the peak radial pressure on the coil (the radial force divided by the coil area at the inner radius). This file was used to generate Figure 5. As indicated, there are 22 terms written for each stage.

fort.22 contains dynamic information: the time, the number of the last coil to fire, the position of the rear of the armature, velocity, acceleration, the armature current, the current in the last coil to fire, and the peak armature temperature. This file was used to generate Figure 3.

fort.23 contains the energy accounting: the time, the position, the resistive energy lost in the coils, the energy in the switched capacitors, the self- and mutual inductive energies, the resistive energy lost in the armature, the kinetic energy, the energy left behind in coils dropped from the tail, the sum of all the above, and the time integrated input power, IV. As an accuracy check, the last two should be equal. This file was used to generate Figure 4.

fort.26 contains electrical information: the time, the position, the numbers of the first and last coils in the tail, and a list of the currents Ia or Ip, Ib, and the capacitor voltages. For the armature elements, Ib and the voltage are zero.

fort.27 contains mechanical and thermal information: the time, the armature position, the numbers of the first and last coils in the tail, and a list of the axial force, the radial force, and the temperature.

fort.28 contains restart information about the system when the run stopped.

fort.29 is read for restart (irun=1). It is produced by renaming or copying *fort.28*.

fort.30 is used by the postprocessor SLINGPLOT to generate plots of the geometry and magnetic field. This file was used to create Figures 2 and 5.

fort.40 is used by the postprocessor SLINGSTRESS for calculating mechanical stress.

IV. Physical Model, Numerical Methods

a) Circuit Equations

There are three facets involved in modeling a coilgun: electrical, thermal, and dynamic.

Let the coils be indexed by "k", running from 1 to K, and the armature elements be indexed by "n", running from 1 to N. Referring to Figure 1, each coil circuit will have currents Ia, and Ib.

Each armature element will have a current I_p . Extra circuits are identical to armature circuits except that they do not move. The circuit equations for the coils and armature (or extra) are:

$$\sum_{\hat{k}=1}^K \frac{d}{dt}(M_{k\hat{k}}I_{a_k}) + \sum_{n=1}^N \frac{d}{dt}(M_{kn}I_{p_n}) + (R_{a_k} + R_{d_k} + R_{c_k})I_{a_k} + L_{a_k} \frac{d}{dt}(I_{b_k}) + R_{a_k}I_{b_k} = V_k$$

$$(L_{a_k} + L_{b_k}) \frac{d}{dt}(I_{b_k}) + (R_{a_k} + R_{b_k})I_{b_k} + L_{a_k} \frac{d}{dt}(I_{a_k}) + R_{a_k}I_{a_k} = V_k$$

$$V_k(t) = V_k(0) - \frac{1}{C_{k0}} \int_0^t (I_{a_k} + I_{b_k}) dt, \quad \text{where all three equation are for } k=1, K.$$

$$\sum_{\hat{n}=1}^N \frac{d}{dt}(M_{n\hat{n}}I_{p_{\hat{n}}}) + \sum_{k=1}^K \frac{d}{dt}(M_{nk}I_{a_k}) + R_{c_n}I_{p_n} = 0 \quad \text{for } n=1, N$$

The off-diagonal terms of the matrices M are the mutual inductances. The diagonal terms, the self-inductances, are given by $M_{kk} = L_{a_k} + L_{d_k} + L_{c_k}$ for the coils and $M_{nn} = L_{p_n}$ for the armature. These equations are put into a centered finite difference form and the resulting linear system for the currents is solved using an LU decomposition algorithm [7].

Temperature dependent resistivities, $\eta(T)$, and specific heats, $\kappa(T)$ are computed for each element. An element's resistance is its resistivity times the mean circumference times the number of turns squared divided by the cross-sectional area. At each time step, every element's temperature is updated according to:

$$m\kappa(T) \frac{dT}{dt} = I^2 R(T) = I^2 N^2 \eta(T) \frac{\text{Circumf}}{\text{Area}} \quad \text{where } m \text{ is the element's mass.}$$

The dependence of η and κ on T are piecewise linear curve fits to data obtained mostly from the Handbook of Thermophysical Properties of Solid Materials [8], although other sources have been used. The mass of each element is computed from its density and volume. Densities are provided for each material (the temperature dependence of density is ignored). The code calculates each element's volume.

b) Inductance Calculations

Perhaps the most important, and most demanding, part of the calculation is finding the self inductance for each element and the mutual inductances between pairs of elements. The self inductance of a circular ring with rectangular cross-section is found using formulas and tables from Grover, p. 105-113 [9]. The tables are included in SLINGSHOT as data. The mutual inductance between two circular hoops with rectangular cross-section is found using Lyle's method (Grover p. 12), which replaces each hoop with two circular filaments, as shown in Figure 7a, computes the mutual inductances between pairs of filaments, and sums them to approximate the actual mutual inductance. Lyle's formula determines the placement of the filaments. If the aspect ratio of any element is greater than 5, the code replaces it with two elements, each with half the longer dimension, for the inductance calculation.

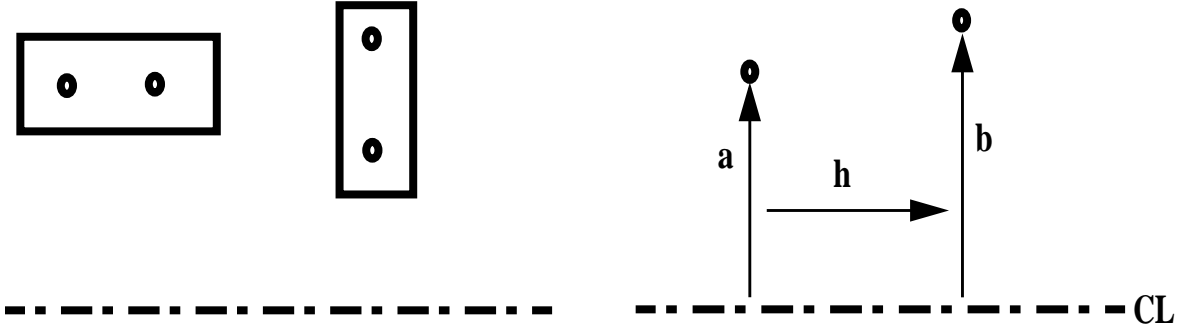


Figure 7. a) Lyle's method of finding mutual inductance by replacing the rectangular hoop by two coaxial circular filaments. b) Two filaments with parameters used in inductance calculation.

The computation of the mutual inductance between two coaxial filaments requires the evaluation of elliptic integrals. Let the two filaments have radii “a” and “b” and be separated by “h”, as shown in Figure 7b. The mutual inductance is

$$M = \mu_0 ab \int_0^\pi \frac{\cos \theta}{\sqrt{h^2 + a^2 + b^2 - 2ab \cos \theta}} d\theta$$

which, after some manipulation, can be written

$$M = \frac{\mu_0 \sqrt{abw}}{\sqrt{2}} \frac{1-w}{1-w} \int_0^\pi \frac{\cos \theta}{\sqrt{1-w \cos \theta}} d\theta \equiv \frac{\mu_0 \sqrt{abw}}{\sqrt{2}} Q_1(w) \quad \text{where } w = 2ab/(h^2 + a^2 + b^2).$$

The function $Q_1(w)$ is evaluated numerically and given in tabular form ($0 < w < 1$). The code interpolates to the actual value of w .

The force in any direction, “x”, between two current carrying elements is given by the product of the currents times the gradient of the mutual inductance:

$$F_x = I_1 I_2 \frac{\partial M_{12}}{\partial x}$$

The axial force and radial forces on the filaments can be obtained from

$$\frac{\partial M}{\partial h} = \frac{\mu_0}{2^{3/2}} \frac{h}{\sqrt{abw}} \left(\frac{w}{1-w} \right)^2 \int_0^\pi \frac{(1-w)^2 \cos \theta}{(1-w \cos \theta)^{3/2}} d\theta \equiv \frac{\mu_0}{2^{3/2}} \frac{h}{\sqrt{abw}} \left(\frac{w}{1-w} \right)^2 Q_2(w) \quad \text{and}$$

$$\frac{\partial M}{\partial a} = \frac{\mu_0}{2^{3/2}} \frac{h}{a \sqrt{abw}} \left(\frac{w}{1-w} \right)^2 \left[(h^2 + b^2) Q_2(w) - ab \int_0^\pi \frac{(1-w)^2 \cos^2 \theta}{(1-w \cos \theta)^{3/2}} d\theta \right] \equiv$$

$$\frac{\mu_0}{2^{3/2}} \frac{h}{a\sqrt{abw}} \left(\frac{w}{1-w} \right)^2 [(h^2 + b^2)Q_2(w) - abQ_3(w)]$$

In addition to $Q_1(w)$, $Q_2(w)$ and $Q_3(w)$ are also tabulated as data (called q, qq, and qqq).

The velocity and position of the armature package is updated at each time step using its total mass (armature plus payload) and the sum of the axial forces on all the armature elements. Both the axial and radial forces are used to compute coil stresses, but only the axial component influences the motion (one hopes!).

Acknowledgments

It is a pleasure to acknowledge the help I received from Bill Cowan, Ron Kaye, and Isaac Shokair in writing and implementing SLINGSHOT. Thanks also to Ron Kaye for reviewing this report, catching several errors, and making many constructive comments.

References

- [1] M. Cowan, M. M. Widner, E. C. Cnare, B. W. Duggin, R. J. Kaye, and J. R. Freeman, "Exploratory Development of the Reconnection Launcher 1986-1990", IEEE Transactions on Magnetism, Vol. 27, No. 1, January, 1991.
- [2] R. J. Kaye, E. L. Brawley, B. W. Duggin, E. C. Cnare, D. C. Rovang, and M. W. Widner, "Design and Performance of a Multi-Stage Cylindrical Reconnection Launcher", IEEE Transactions on Magnetism, Vol. 27, No. 1, January, 1991.
- [3] R. J. Kaye, E. C. Cnare, M. Cowan, B. W. Duggin, R. J. Lipinski, and B. M. Marder," Design and Performance of Sandia's Contactless Coilgun for 50 mm Projectiles", IEEE Transactions on Magnetism, Vol. 29, No. 1, January, 1993.
- [4] B. M. Marder,"A Coilgun Design Primer", IEEE Transactions on Magnetism, Vol. 29, No. 1, January, 1993
- [5] I. R. Shokair, M. Cowan, R. J. Kaye, and B. M. Marder, "Performance of an Induction Coil Launcher", IEEE Transactions on Magnetism, Vol. 31, No. 1, January, 1995.
- [6] M. M. Widner, "WARP-10: A Numerical Simulation Model for the Cylindrical Reconnection Launcher", IEEE Transactions on Magnetism, Vol. 27, No. 1, January, 1991.
- [7] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Numerical Recipes, Cambridge University Press, 1986.
- [8] A Goldsmith, T. E. Waterman, H. J. Hirschhorn, and the Armour Research Foundation, Handbook of Thermophysical Properties of Solid Materials, Volume I: Elements, The Macmillan Company, 1961.
- [9] F. W. Grover, Inductance Calculations, Dover Publications, Inc., 1946.

Distribution

1	MS 1153	Jeff Alexander, 15331
5	MS 1182	Ron Kaye, 15335
1	MS 1182	Bob Turman, 15335
1	MS 1182	Steve Shope, 15335
1	MS 1193	Dean Rovang, 1645
1	MS 1186	Tom Mehlhorn, 1674
1	MS 9103	Isaac Shokair, 8120
10	MS 1186	Barry Marder, 1674
1	MS9018	Central Technical Files, 8945-1
2	MS0899	Technical Library, 9616
1	MS0612	Review and Approval Desk for DOE/OSTI, 9612